Network Working Group
Request for Comments: 726
NIC: 39237

Jon Postel & Dave Crocker
SRI-ARC UC Irvine
8 March 1977

# Remote Controlled Transmsssion and Echoing Telnet Option

1

## 1. Command name and code: 2

RCTE 7 2a

## 2. Command meanings: 3

IAC WILL RCTE 3a

The sender of this command REQUESTS or AGREES to use the RCTE option, and will send instructions for controlling the other side's terminal printer. 3a1

IAC WON'T RCTE 3b

The sender of this option REFUSES to send instructions for controlling the other side's terminal printer. 3b1

IAC DO RCTE 3c

The sender REQUEST or AGREES to have the other side (sender of WILL RCTE) issue commands which will control his (sender of the DO) output to the terminal printer. 3c1

IAC DON'T RCTE 3d

The sender of this command REFUSES to allow the other side to control his (sender of DON'T) terminal printer. 3d1

IAC SB RCTE <cmd> [BC1 BC2] [TC1 TC2] IAC SE 3e

where: 3e1

<cmd> is one 8-bit byte having the following flags (bits are counted from the right): 3e1a

| Bit | Meaning | 3e1b |
|---|---|---|
| 0 | 0 = Ignore all other bits in this byte and repeat the last <cmd> that was sent. Equals a 'continue what you have been doing'.<br>1 = Perform actions as indicated by other bits in this byte. | 3e1c |
| 1 | 0 = Print (echo) break character<br>1 = Skip (don't echo) break character | 3e1d |
| 2 | 0 = Print (echo) text up to break character<br>1 = Skip (don't echo) text up to break character | 3e1e |
| 3 | 0 = Continue using same classes of break characters.<br>1 = The two 8-bit bytes following this byte contain flags for the new break classes. | 3e1f |
| 4 | 0 = Continue using same classes of transmit characters.<br>1 = Reset transmit classes according to the two bytes following 1) the break classes bytes, if the break classes are also being reset, or 2) this byte, if the break classes are NOT also being reset. | 3e1g |

Value (decimal) of the <cmd> byte and its meaning: 3e1h

0 = Continue what you have been doing 3e1i

Even numbers greater than zero (i.e. numbers with the right most bit off) are in error and should be interpreted as equal to zero. When the <cmd> is an even number greater than zero, classes bytes TC1 & TC2 and/or BC1 & BC2 must not be sent. 3e1j

1 = Print (echo) up to AND INCLUDING break character 3e1k

3 = Print up to break character and SKIP (don't echo) break character 3e1l

5 = Skip text (don't echo) up to break character, but PRINT break character 3e1m

7 = Skip up to and including break character 3e1n

Add one of the previous non-zero values to one of the following values, to get the total decimal value for

the byte (Note that classes may not be reset without also resetting the printing action; so an odd number is guaranteed): 3e1o

8 = Set break classes (using the next two bytes [BC1 BC2]) 3e1p

16 = Set transmission classes (using the next two bytes [TC1 TC2]) 3e1q

24 = Set break classes (using the next two bytes [BC1 BC2]) and the transmission classes (using the two bytes after that [TC1 TC2]). 3e1r

Sub-commands (IAC SB RCTE...) are only sent by the controlling host and, in addition to other functions, functionally replace the Go-Ahead (IAC GA) Telnet feature. RCTE also functionally replaces the Echo (IAC ECHO) Telnet option. That is the Suppress Go-Ahead option should be in force and the Echo option should not be in force while the RCTE option is in use. The echo mode on terminating use of the RCTE option should be the default state, that is DON'T ECHO, WON'T ECHO. 3e2

Classes for break and transmission (the right-most bit of the second byte (TC2 or BC2) represents class 1; the left-most bit of the first byte (TC1 or BC1) represents the currently undefined class 16: 3e3

1: Upper-Case Letter (A-Z) 3e3a

2: Lower-case Letters (a-z) 3e3b

3: Numbers (0-9) 3e3c

4: Format Effectors (<BS> <CR> <LF> <FF> <HT> <VT>) 3e3d

The sequence <cr><lf> counts as one character when processed as the Telnet end of line, and is a single break character when class 4 is set. The sequence <cr><nul> counts as one character and is a break character if and only if <cr> is a break character (i.e. class 4 is set).

5: Non-format Effector Control Characters including <DEL> and <ESC> 3e3e

6: . , ; : ? ! 3e3f

7: { [ ( < > ) ] } 3e3g

8: ' "  / \ % @ $ & # + - * = ^ _ | ~ 3e3h

9: <Space> 3e3i

And Telnet commands (IAC . . .) sent by the user are always to have the effect of a break character. That is, every instance of an IAC is to be treated as a break character, except the sequence IAC IAC. 3e3j

The representation to be displayed when printing is called for is the obvious one for the visible characters (classes 1, 2, 3, 6, 7, and 8). Space (class 9) is represented by a blank space. The format effectors (class 4) by their format effect. The non-format effector controls (class 5) print nothing (no space). 3e4

Initially no break classes or transmission classes are in effect. 3e5

Please note that if all the bits are set in a Telnet subcommand argument byte such as TC2 or BC2 then that byte must be preceeded by an <IAC> flag byte. This is the common convention of doubling the escape character to use its value as data. 3e6

Sub-commands (IAC SB RCTE...) are refered to as "break reset commands". 3e7

3. Default: 4

WON'T RCTE -- DON'T RCTE 4a

Neither host asserts special control over the other host's terminal printer. 4a1

4. Motivation for the option: 5

RFC's 1, 5 and 51 discuss Network and process efficiency and smoothness. 5a

RFC 357, by John Davidson, introduces the problem of echoing delay that occurs when a remote user accesses a full-duplex host, thru a satellite link. In order to save the many thousands of miles of transit time for each echoed character, while still permitting full server responsiveness and clean terminal output, an echo control

similar to that used by some time-sharing systems is suggested for the entire Network. 5b

In effect, the option described in this document involves making a using host carefully regulate the local terminal printer according to explicit instructions from the remote (serving) host. 5b1

An important additional issue is efficient Network transmission. Implementation of the Davidson Echoing Scheme will eliminate almost all server-to-user echoing. 5c

The option described in this document also requests using hosts to buffer a terminal's input to the serving host until it forms a useful unit (with "useful unit" delimited by break or transmission characters as described below). Therefore, fewer messages are sent on the user-to-server path. 5c1

N.B.: This option is only intended for use with full-duplex hosts. The Go-Ahead Telnet feature is completely adequate for half-duplex server hosts. Also, RCTE should be used in place of the ECHO Telnet option. That is the Suppress Go-Ahead option should be in force and the Echo option should not be in force while the RCTE option is in use. 5d

## 5. Explicit description of control mechanism: 6

### User Terminal Printing Action & Control Procedure 6a

Negotiate the use of the RCTE option. Once the option is in force the user Telnet follows the following procedure. 6a1

1) Read an item from the network. 6a2

If the item is data, then print it and go to 1. 6a2a

If the item is a command, then set the classes and go to 2. 6a2b

2) If the terminal input buffer is empty, then go to 3, else go to 4. 6a3

3) Wait for an item to appear either from the terminal or from the network. 6a4

If an item appears from the terminal, then go to 4. 6a4a

If a data item appears from the network, then print it and go to 3. 6a4b

If a command appears from the network, then an error has occured. 6a4c

4) Read an item from the terminal input buffer. 6a5

If the item is not a break, then print/skip it and go to 2. 6a5a

If the item is a break, then print/skip it and go to 1. 6a5b

Note: Output from the server host may occur at any time, such "spontaneous output" is printed in step 3. 6a6

Explanation: 6b

Both Hosts agree to use the RCTE option. After that, the using host (IAC DO RCTE) merely acts upon the controlling (serving) host's commands and does not issue any RCTE commands unless and until it (using host) decides to stop allowing use of the option (by sending IAC DON'T RCTE). 6b1

1) The using host is synchronized with the server by initially and when ever it returns to step 1 suspending terminal echo printing until it receives a command from the server. 6b2

The server may send either output to the terminal printer or a command, and usually sends a both. 6b3

The server may send output to the terminal printer either in response to user input or spontaneously. In the former case, the output is processed in step 1. In the latter case, the output is processed in step 3. 6b4

Server sends an RCTE command. The command may redefine break and transmission classes, action to be performed on break characters, and action to be performed on text. Each of these independent functions is controlled by separate bits in the <cmd> byte. 6b5

A transmission character is one which RECOMMENDS that the using host transmit all text accumulated up to and including its occurrence. (For network efficiency, using hosts are DISCOURAGED (but not prohibited) from sending before the occurrence of a transmission character, as defined at the moment the character is typed). 6b5a

If the transmission classes bit (bit 4) is on, the two bytes following the two break classes bytes (or immediately following the <cmd> byte, if the break classes bit is not on) will indicate what classes are to be enabled.

If the bit is OFF, the transmission classes remain unchanged. When the RCTE option is first initiated, NO CLASSES are in effect. That is, no character will be considered a transmission character. (As if both TC1 and TC2 are zero.)

A break character REQUIRES that the using host

transmit all text accumulated up to and including its occurrence and also causes the using host to stop its print/discard action upon the user's input text, until directed to do otherwise by another IAC SB RCTE <cmd> IAC SE command from the serving host. Break characters therefore define printing units. "Break character" as used in this document does NOT mean Telnet Break character. 6b5b

If the break classes bit (bit 3) is on, the two bytes following <cmd> will indicate what classes are to be enabled. There are currently nine (9) classes defined, with room for expansion.

If the bit is OFF, the break classes remain unchanged. When the RCTE option is initiated, NO CLASSES are to be in effect. That is, no transmission will take place in the user to server direction until the first break reset command is received by the user from the server.

The list of character classes, used to define break and transmission classes are listed at the end of this document, in the Tables Section. 6b5c

Because break characters are special, the print/discard action that should be performed upon them is not always the same as should be performed upon the rest of the input text. 6b5d

For example, while typing a filename to TENEX, I want the text of the filename to be printed (echoed); but I do not want the <escape> (if I use the name completion feature) to be printed.

If bit 1 is ON the break character is NOT to be printed.

A separate bit (bit 2) signals whether or not the text itself should be printed (echoed) to the terminal. If bit 2 = 0, then the text IS to be printed. 6b5e

Yet another bit (bit 0 - right-most bit) signals whether or not any of the other bits of the command should be checked. If this bit is OFF, then the command should be interpreted to mean "continue whatever echoing strategy you have been following, using the same break and transmission classes." 6b5f

2) The user Telnet now checks the terminal input buffer, if it contains data it is processed in step 4, otherwise the user Telnet waits in step 3 for further developments. 6b6

3) The user Telnet waits until either the human user enters some data in which case Telnet proceeeds to step 4, or an item is received from the network. If the item from the network is data it is spontaneous output and is printed, Telnet then continues to wait. If the item from the network is a command then an error has occured. In this case the user Telnet may attempt to resynchronize the use of RCTE as indicated below. 6b7

4) Items from the terminal are processed with printing controlled by the settings of the latest break reset command. When a break character is processed, the cycle of control is complete and action re-commences at step 1. 6b8

Input from the terminal is (hopefully) buffered into units ending with a transmission or break character; and echoing of input text is suspended after the occurrence of a break character and until receipt of a break reset command from the serving host. The most recent break reset command determines the break actions. 6b9

In summary, what is required is that for every break character sent in the user to server direction there be a break reset command sent in the server to user direction. The user host initially has no knowledge of which characters are break characters and so starts in a state that assumes that there are no break characters and also that no echoing is to be provided. The server host is expected to send a break reset command to establish the break classes and the echoing mode before it receives any data from the user. 6b10

Synchronization and Resynchronization: 6c

The serving and using hosts must carefully synchronize break reset commands with the transmission of break characters. Except at the beginning of an interaction, the serving host may only send a break reset command in response to the Using host's having sent a break character as defined at that time. This should establish a one-to-one correspondence between them. (A <cmd> value of zero, in this context, is interpreted as

a break classes reset to the same class(es) as before.) The break reset command may be preceded by terminal output. 6c1

The re-synchronization of the break characters and the break reset commands is done via the exchange of the Telnet signal Abort Output (AO) in the server to user direction and the SYNCH in the user to server direction. 6c2

Suppose the server wants to resynchronize the break characters and the break reset commands. 6c3

a. The server should be sure all output to the terminal has been printed by using, for example, the Timing Mark Option. 6c3a

b. The server sends the AO signal. 6c3b

c. The user receives the AO signal. The user flushes all user to server data wheather it has been echoed or not. The user sends a SYNCH to the server. [The SYNCH consists of the Telnet Data Mark (DM) and the host-to-host interrupt (INS).] The user now enters the initial state at step 1. 6c3c

d. The server receives the SYNCH and flushes any data preceeding the DM (as always). The server now sends a break reset command. (Actually the break reset command could be sent at any time following the AO.) 6c3d

Suppose the user wants to resynchronize the break characters and the break reset commands. 6c4

a. The user should discard all user to server data wheather it has been echoed or not. 6c4a

b. The user sends the AO signal. The user now enters the algorithm at step 1. 6c4b

c. The server receives the AO signal. The server discards all data buffered but not yet sent to the user. The server sends a SYNCH to the user. The server sends a break reset command to the user. 6c4c

Notes and Comments: 6d

Even-numbered commands, greater than zero, are in error, since they will have the low-order bit off. The command should be interpreted as equal to zero, which means that any classes reset bytes ([TC1 TC2] [BC1 BC2]) will be in error. (The IAC SE, at the end of the command, eliminates any parsing problems due to this error.) 6d1

Serving hosts will generally instruct using hosts not to echo break characters, even though it might be alright to echo most break characters. For example, <cr> is usually a safe character to echo but <esc> is not. TENEX Exec is willing to accept either, during filename specification. Therefore, the using host must be instructed not to echo any break characters. 6d2

This is generally a tolerable problem, since the serving host has to send an RCTE command at this point, anyhow. Adding an echo for the break character to the message will not cause any extra network traffic. 6d2a

The RCTE Option entails a rather large overhead. In a true character-at-a-time situation, this overhead is not justified. But on the average, it should result in significant savings, both in network traffic and host wake-ups. 6d3

Buffering Problems and Transmission vs. Printing Constraints: 6d4

There are NO mandatory transmission constraints. The using host is allowed to send a character a time, though this would be a waste of RCTE. The transmission classes commands are GUIDELINES, so deviating from them, as when the user's buffer gets full, is allowed. 6d4a

Additionally, the using host may send a break class character, without knowing that it is one (as with type-ahead). 6d4b

If the user implementation is clever it may send the user entered data to the server before it is actually needed. This type ahead data may contain break characters.

Assume that only space is a break character (that is the last break reset command specified print up to and including the break characters and set the break classes to class 9). Suppose the user had typed "abc<space>def<esc>ghi<cr>". The user side RCTE could send it all to the server, but it could print only "abc<space>", and would have to buffer "def<esc>ghi<cr>" at least until a break reset command was received from the server. That break reset command could change the break ckasses requiring rescanning the buffered string.

For example suppose the break reset command set the break characters to class 5 and the action to print up to but not including the break character. The user RCTE could then print "def" and discard the <esc>, but whould have to continue to buffer the "ghi<cr>".

The problem with buffering occurs when printing on the user's terminal must be suspended, after the user has typed a currently valid break character and until a break reset command is received from the serving host. During this time, the user may be typing merrily along. The text being typed may be SENT, but may not yet be PRINTED. 6d4c

The more common problem of filling the transmission buffer, while awaiting a host to host allocate from the serving host, may also occur, but this problem is well known to implementors and in no way special to RCTE. 6d4d

In any case, when the buffer does fill and further text typed by the user will be lost, the user should be notified (perhaps by ringing the terminal bell). 6d4e

Text should be buffered by the using host until the user types a character which belongs to the transmission class in force at the moment the character is typed. 6d5

Transmission class reset commands may be sent by the serving host at any time. If they are frequently sent separate from break class reset commands, it will probably be better to exit from RCTE and enter regular character at a time transmission. 6d6

It is not immediately clear what the using host should

do with currently buffered text, when a transmission classes reset command is received. The buffering is according to the previous transmission classes scheme. 6d7

The using host clearly should not simply wait until a transmission character (according to the new scheme) is typed. 6d7a

Either the buffered text should be rescanned, under the new scheme; 6d7b

Or the buffered text should simply be sent as a group. This is the simpler approach, and probably quite adequate. 6d7c

It is possible to define NO BREAK CHARACTERS except Telnet commands (IAC ...). This seems undesirable and should not be done. 6d8

If this situation were to occur the using host should send a Telnet command to allow the server to know when he may reset the break classes, but the mechanism is awkward and this case should be avoided. 6d8a

## 6. Sample Interaction: 7

"S:" is sent from serving (WILL RCTE) host to using host.
"U:" is sent from using (DO RCTE) host to serving host.
"T:" is entered by the terminal user.
"P:" is printed on the terminal.

Text surrounded by square brackets ([]) is commentary. Text surrounded by angle brackets (<>) is to be taken as a single unit. E.g., carriage return is <cr>, and the decimal value 27 is represented <27>. 7a

The following interaction shows a logon to a Tenex, initiation of the DED editor, insertion of some text and the return to the Exec level. 7b

An attempt has been made to give some flavor of the asynchrony of network I/O and the user's terminal input. Many other possible combinations, using the same set of actions listed below, could be devised. The actual order of events will depend upon network and hosts' load and the user's typing speed. 7b1

we assume that the user's Telnet is also in an "insert linefeed" mode. That is, whenever the user types carriage

return <cr> the user Telnet sends both carriage return and linefeed <cr><lf> (the Telnet end of line signal). When space character occurs at the end of a line in the example description it is shown explicitly by <sp> to avoid confusion. Other uses of the space character are not so marked to avoid destroying the readability of the example. 7c

A Telnet connection has already been opened, but the TENEX prompt has not yet been issued. The hosts first discuss using the RCTE option: 7d

S: <IAC><WILL><RCTE> 7d1

U: <IAC><DO><RCTE> 7d2

S: TENEX 1.31.18, TENEX EXEC 1.50.2<cr><lf>@
<IAC><SB><RCTE><11><1><24><IAC><SE> 7d3

[Print the herald and echo input text up to a break character, but do not echo the break character. Classes 4 (Format Effectors), 5 (Non-format Effector Controls and <DEL>), and 9 (<sp>) act as break characters.] 7d3a

P: TENEX 1.31.18, TENEX EXEC 1.50.2<cr><lf>@ 7d4

T: LOGIN ARPA<cr> 7d5

P: LOGIN 7d6

U: LOGIN<sp> 7d7

U: ARPA<cr><lf> 7d8

S: <sp><IAC><SB><RCTE><0><IAC>SE> 7d9

P: <sp>ARPA 7d10

S: <cr><lf>(PASSWORD): <IAC><SB><RCTE><7><IAC><SE> 7d11

P: <cr><lf>(PASSWORD):<sp> 7d12

T: WASHINGTON 1000<cr> 7d13

[The password "WASHINGTON" is not echoed. Printing of "1000<cr>" is withheld] 7d13a

U: WASHINGTON<sp> 7d14

U: 1000<cr><lf> 7d15

S: <sp><IAC><SB><RCTE><3><IAC><SE> 7d16

S: <cr><lf>JOB 17 ON TTY41 7-JUN-73 14:13<cr><lf>@
<IAC><SB><RCTE><0><IAC><SE> 7d17

P: <sp>1000 7d18

[Printing is slow at this point; so the account number is not printed as soon as the server's command for it is received.] 7d18a

P: <cr><lf>JOB 17 ON TTY41 7-JUN-73 14:13<cr><lf>@ 7d19

T: DED<esc><cr> 7d20

P: DED 7d21

U: DED<esc> 7d22

S: .SAV;1<IAC><SB><RCTE><0><IAC><SE> 7d23

P: .SAV;1 7d24

U: <cr><lf> 7d25

S: <cr><lf><lf>DED 3/14/73 DRO,KRK<cr><lf>:
<IAC><SB><RCTE><15><1><IAC><255><IAC><SE> 7d26

[The program is started and the DED prompt ":" is sent. At the command level, DED responds to every character. The server sets the break classes to all classes.] 7d26a

P: <cr><lf><lf>DED 3/14/73 DRO,KRK<cr><lf>: 7d27

T: IThis is a test line.<cr>This is another test
line.<^Z>Q 7d28

["I" means Insert Text. The text follows, terminated by a Control-Z. The "Q" instructs DED to Quit.] 7d28a

U: I 7d29

U: This is a test line.<cr><lf> 7d30

S: I<cr><lf>*<IAC><SB><RCTE><11><0><24><IAC><SE> 7d31

[DED prompts the user, during text input, with an asterisk at the beginning of every line. The server sets the break classes to classes 4 and 5, the format effectors and the non-format effector controls.] 7d31a

P: I<cr><lf>*This is a test line. 7d32

S: <cr><lf>*<IAC><SB><RCTE><0><IAC><SE> 7d33

P: <cr><lf>*This is another test line. 7d34

U: This is another test line.<^Z> 7d35

U: Q 7d36

[Note that the "Q" will not immediately be printed on the terminal, since it must wait for authorization.] 7d36a

S: ^Z<cr><lf>:<IAC><SB><RCTE><15><1><IAC><255><IAC><SE> 7d37

[The returned "^Z" is two characters, not the ASCII Control-Z or <sub>.] 7d37a

S: Q<cr><lf>@<IAC><SB><RCTE><11><1><24><IAC><SE> 7d38

P: Q<cr><lf>@ 7d39

And the user is returned to the Exec level. 7d40